REMARKS/ARGUMENTS

In response to the Office Action mailed December 28, 2005, finally rejecting claims, Applicants request reconsideration in view of the proposed Amendment and following remarks. In this Amendment it is proposed to cancel claims 8-10 and 13, leaving claims 6, 7, 11, 12, and 14-20 pending.

The Abstract was objected to but it is not apparent what the basis of the objection was. The Abstract did not contain any words such as "means" or "said" and was not unduly lengthy. Nevertheless, to advance the prosecution, a substitute Abstract is supplied.

Claims 8 and 10 were rejected as indefinite as employing a trademark. The cancellation of claims 8-10 and 13 eliminates the basis for the claim rejection.

All pending claims were rejected as unpatentable over DeLorme et al. (U.S. Patent 6,321,158, hereinafter DeLorme) in view of an excerpt from a book entitled *Inside the Java Virtual Machine* (hereinafter JVM). This rejection is respectfully traversed as to the claims still pending.

Applicants do not fundamentally disagree with the Examiner's description of DeLorme. In addition to a platform block, Applicants agree that DeLorme discloses a navigation apparatus including an optional application processing block, an interface processing block, and a navigation application processing block. All of these three blocks in DeLorme are written using what JVM refers to as a Native language, for example, Windows. Thus, as understood by those of skill in the art, the optional application processing block is dependent upon the platform block.

Figure 9 of the patent application illustrates an embodiment of the invention in a block diagram that is of assistance in understanding, both with respect to claim 6, the sole pending independent claim, and DeLorme, the arrangement of these blocks. As described in the patent application, the platform block comprises, in the embodiment of Figure 9, the navigation operating system, a device driver, and the navigation hardware, elements 42, 43, and 41. The navigation application module 46 encompassing the navigation application processing block of claim 6 is a complicated program providing navigation services. That navigation application processing block can draw upon information from

an optional application processing block that adds optional services regarding navigation. In the embodiment of Figure 9, the distribution application module is an example of such an optional application processing block. For the described embodiment of the patent application, the distribution application module causes the navigation information that is generated to relate to the delivery of goods at various stops made by a vehicle on a particular route.

The optional application processing block and the navigation application processing block communicate with each other through an interface module processing block encompassing the interface module 112 of the embodiment of Figure 9. In the invention, the interface processing block is executed on a virtual platform and is, therefore, independent of the platform block. This description in claim 6 encompasses, for example, the execution of the interface processing block on a Java virtual machine, element 44 in Figure 9, as indicated in that figure and as described in the patent application. In the embodiment of the invention described in the patent application not only is the interface module 112 executed on a virtual platform, but also the optional application processing block uses the Java language, not a Native language, such as Windows.

As well known to those of skill in the art, navigation application modules have been available in Native languages, such as C or C++, for some time. Because these programs are extraordinarily complicated and include an enormous number of steps, writing such a program in Java is too burdensome to be reasonably undertaken. Thus, as in DeLorme, in the invention, the navigation application processing block represents a block operating in a Native language.

As at least implicitly acknowledged in the rejection, in DeLorme neither the optional application processing block nor the interface module is executed on a virtual platform. Rather, those blocks are executed in a Native language. Therefore, there is no independence of these blocks from the platform block. Thus, the reliance upon JVM is based upon an assertion that it would have been obvious to modify DeLorme in a way that produces the claimed invention by basing at least the interface block on Java. Applicants disagree with the asserted hypothetical modification of DeLorme.

In DeLorme, a function of the interface processing block is called by the optional application processing block in order to use the functions of the navigation application processing block. To achieve that result, based upon the description in JVM, particularly at pages 32, 33, 126, and 127, one of skill in the art must proceed with care to avoid the difficulties outlined at those two pages of JVM. Because of the limitations set out in JVM, that person of skill in the art would be led to use an interface processing block written in a Native language so that functions of the interface processing block could be directly called from the optional application processing block that is in the Java language. In other words, the structure that would be produced by the hypothesized modification of DeLorme might include an optional processing block in the Java language. However, the interface processing block would remain in a Native language in order to provide direct access to the navigation application processing block. As previously explained, that navigation processing block, most advantageously, is maintained in all systems in a Native language. Thus, platform independence would not be achieved. In other words, it is counterintuitive, and, therefore, not obvious, to provide the interface processing block, as in the invention, to be executed on a virtual platform because of the problems in doing so that JVM warns about. See page 32 of JVM stating that the "most important rule to follow when you are writing a platform-independent Java program is: don't directly or indirectly invoke any Native methods that aren't part of the Java API".

To emphasize the differences between the hypothetically modified DeLorme and the invention, in the modified DeLorme, functions of the interface processing block, which is written in a Native language, are directly called from the optional application processing block, which is written in Java. Therefore, the optional application processing block is dependent upon the platform. By contrast, in the invention, since the optional application processing block, which is written in Java, does not directly call a program written in a Native language, but calls through the interface processing block which is executed on a virtual platform, there is no dependence on the platform block. On this basis, reconsideration and withdrawal of the rejection as to claim 6, the sole pending independent claim, is respectfully requested. Since that claim 6 is, for the reasons

supplied, patentable over the asserted combination of prior art publications, so are the remaining pending claims, all of which depend directly or indirectly from claim 6.

Since the foregoing Amendment merely cancels claims and attends to a formality issue, it cannot raise any new questions that require a new search. Accordingly, even if the rejection is maintained, the Amendment must be entered for the purposes of appeal.

Reconsideration and allowance of all claims new pending are earnestly solicited.

Respectfully submitted,

Jeffrey A. Wyand, Reg. No. 29,458
LEYDIG, VOIT & MAYER
700 Thirteenth Street, N.W., Suite 300
Washington, DC 20005-3960
(202) 737-6770 (telephone)
(202) 737-6776 (facsimile)

Date: March 21, 2006
Jaw:ves